This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1-14 (Canceled)

15. (Currently Amended) A method for a secure processor to instantiate and authenticate a secure application thereon by way of a security kernel, the method comprising:

entering a preferred mode where a security key of the processor is accessible;

instantiating and running a security kernel, the security kernel:

accessing the security key;

applying the accessed security key to decrypt at least one encrypted key for the application;

storing the decrypted key(s) in a location where the application will expect the key(s) to be found; and

authenticating the application on the processor; and

entering a normal mode from the preferred mode after the security kernel authenticates the application, where the security key is not accessible;

wherein the security kernel allows the processor to be trusted to keep hidden the key(s) of the application, and

wherein the security kernel employs the accessed security key during the preferred mode to authenticate / verify the application prior to instantiation thereof.

16. (Original) The method of claim 15 wherein entering the preferred mode comprises entering the preferred mode upon a CPU reset.

17. (Original) The method of claim 15 further comprising erasing data in a cache of the processor prior to instantiating the security kernel.

18. (Original) The method of claim 15 further comprising erasing data in a cache of the processor after entering normal mode.

19. (Canceled)

20. (Currently Amended) The method of claim [[19]] 15 wherein the security kernel performs a hash / MAC (message authentication code) over at least a portion of the application and then compares the hash / MAC to a hash / MAC corresponding to the application.

21. (Original) The method of claim 15 wherein the security key of the processor is a symmetric key and the application is instantiated from a code image including a main body and a header including:

KCPU (KMAN)	KMAN encrypted according to KCPU
KMAN (KCODE)	KCODE encrypted according to KMAN

where KCPU is the security key, KMAN is a device key of the portable device independent of the security key, and KCODE is the secret of the application, and

wherein the security kernel applying the accessed security key to decrypt at least one encrypted key for the application comprises:

applying KCPU to KCPU (KMAN) to produce KMAN; and

applying KMAN to KMAN (KCODE) to produce KCODE.

22. (Original) The method of claim 21 wherein the security key of the processor is a symmetric key and the application is instantiated from a code image including a main body and a header including:

KCPU (KMAN)	KMAN encrypted according to KCPU
MAC (main body, KMAN)	message authentication code of the main body under KMAN
KMAN (KCODE)	KCODE encrypted according to KMAN

where KCPU is the security key, KMAN is a device key of the portable device independent of the security key, and KCODE is the secret of the application, and

wherein the security kernel applying the accessed security key to decrypt at least one encrypted key for the application comprises:

applying KCPU to KCPU (KMAN) to produce KMAN;

computing MAC (main body, KMAN);

comparing the computed MAC to MAC (main body, KMAN) from the header to determine if the code image has been changed; and

if the MACs match, applying KMAN to KMAN (KCODE) to produce KCODE.

23. (Original) The method of claim 15 wherein the security key of the processor is a private key of a public key - private key pair and the application is instantiated from a code image including a main body and a header including:

public key (KCODE)	KCODE encrypted according to the public key

where KCODE is the secret of the application, and

wherein the security kernel applying the accessed security key to decrypt at least one encrypted key for the application comprises applying the security key as the private key to public key (KCODE) to produce KCODE.

24. (Original) The method of claim 23 wherein the security key of the processor is a private key of a public key - private key pair and the application is instantiated from a code image including a main body and a header including:

public key (HASH (main body), KCODE)	Hash of the main body and KCODE, both encrypted according to the public key

where KCODE is the secret of the application, and

wherein the security kernel applying the accessed security key to decrypt at least one encrypted key for the application comprises:

computing HASH (main body);

applying the private key to public key (HASH (main body), KCODE) to produce HASH (main body) and KCODE;

comparing the computed HASH to the produced HASH to determine if the code image has been changed;; and

if the HASHs match, employing the produced KCODE as appropriate.

25. (Original) A method for a secure processor to instantiate one of a plurality of available secure applications thereon by way of a security kernel, the method comprising:

setting a chooser value to a value corresponding to a chooser application upon power-up;

entering a preferred mode upon a power-up CPU reset and instantiating the security kernel, the security kernel determining that the chooser value corresponds to the chooser application and therefore authenticating same, the chooser application being instantiated;

entering a normal mode after the chooser application is instantiated and leaving same to run, the chooser application presenting the plurality of available applications for selection by a user;

receiving a selection of one of the presented applications to be instantiated;

setting the chooser value to a value corresponding to the selected application;

entering a preferred mode upon an executed CPU reset and instantiating the security kernel, the security kernel determining that the chooser value corresponds to the selected application and therefore authenticating same, the selected application being instantiated;

entering a normal mode after the selected application is instantiated and leaving same to run;

wherein the security kernel allows the processor to be trusted to keep hidden a secret of the chooser application and a secret of the selected application.

26. (Original) The method of claim 25 further comprising setting the chooser value to the value corresponding to the chooser application upon the selected application being authenticated by the security kernel, wherein upon execution of a CPU reset, the security kernel determines that the chooser value corresponds to the chooser application 72c and therefore authenticates same.

27. (Original) The method of claim 25 further comprising storing the chooser value in a memory location not affected by a CPU reset so that the stored chooser value is available after same.

28-30 (Canceled)

31. (Currently Amended) A computer-readable medium having stored thereon computer-executable instructions implementing a method for a secure processor to instantiate a secure application thereon by way of a security kernel, the method comprising:

entering a preferred mode where a security key of the processor is accessible;

instantiating and running a security kernel, the security kernel:

accessing the security key;

applying the accessed security key to decrypt at least one encrypted key for the application;

storing the decrypted key(s) in a location where the application will expect the key(s) to be found; and

authenticating the application on the processor; and

entering a normal mode from the preferred mode after the security kernel authenticates the application, where the security key is not accessible;

wherein the security kernel allows the processor to be trusted to keep hidden the key(s) of the application, and

wherein the security kernel employs the accessed security key during the preferred mode to authenticate / verify the application prior to instantiation thereof.

32. (Original) The medium of claim 31 wherein entering the preferred mode comprises entering the preferred mode upon a CPU reset.

33. (Original) The medium of claim 31 wherein the method further comprises erasing data in a cache of the processor prior to instantiating the security kernel.

34. (Original) The medium of claim 31 wherein the method further comprises erasing data in a cache of the processor after entering normal mode.

35. (Canceled)

36. (Currently Amended) The medium of claim [[35]] 31 wherein the security kernel performs a hash / MAC (message authentication code) over at least a portion

of the application and then compares the hash / MAC to a hash / MAC corresponding to the application.

37. (Original) The medium of claim 31 wherein the security key of the processor is a symmetric key and the application is instantiated from a code image including a main body and a header including:

KCPU (KMAN)	KMAN encrypted according to KCPU
KMAN (KCODE)	KCODE encrypted according to KMAN

where KCPU is the security key, KMAN is a device key of the portable device independent of the security key, and KCODE is the secret of the application, and

wherein the security kernel applying the accessed security key to decrypt at least one encrypted key for the application comprises:

applying KCPU to KCPU (KMAN) to produce KMAN; and

applying KMAN to KMAN (KCODE) to produce KCODE.

38. (Original) The medium of claim 37 wherein the security key of the processor is a symmetric key and the application is instantiated from a code image including a main body and a header including:

KCPU (KMAN)	KMAN encrypted according to KCPU
MAC (main body, KMAN)	message authentication code of the main body under KMAN
KMAN (KCODE)	KCODE encrypted according to KMAN

where KCPU is the security key, KMAN is a device key of the portable device independent of the security key, and KCODE is the secret of the application, and

wherein the security kernel applying the accessed security key to decrypt at least one encrypted key for the application comprises:

applying KCPU to KCPU (KMAN) to produce KMAN;

computing MAC (main body, KMAN);

comparing the computed MAC to MAC (main body, KMAN) from the header to determine if the code image has been changed; and

if the MACs match, applying KMAN to KMAN (KCODE) to produce KCODE.

39. (Original) The medium of claim 31 wherein the security key of the processor is a private key of a public key - private key pair and the application is instantiated from a code image including a main body and a header including:

public key (KCODE)	KCODE encrypted according to the public key

where KCODE is the secret of the application, and

wherein the security kernel applying the accessed security key to decrypt at least one encrypted key for the application comprises applying the security key as the private key to public key (KCODE) to produce KCODE.

40. (Original) The medium of claim 39 wherein the security key of the processor is a private key of a public key - private key pair and the application is instantiated from a code image including a main body and a header including:

public key (HASH (main body), KCODE)	Hash of the main body and KCODE, both

	encrypted according to the public key

where KCODE is the secret of the application, and

wherein the security kernel applying the accessed security key to decrypt at least one encrypted key for the application comprises:

computing HASH (main body);

applying the private key to public key (HASH (main body), KCODE) to produce HASH (main body) and KCODE;

comparing the computed HASH to the produced HASH to determine if the code image has been changed;; and

if the HASHs match, employing the produced KCODE as appropriate.

41. (Original) A computer-readable medium having computer-executable instructions thereon implementing a method for a secure processor to instantiate one of a plurality of available secure applications thereon by way of a security kernel, the method comprising:

setting a chooser value to a value corresponding to a chooser application upon power-up;

entering a preferred mode upon a power-up CPU reset and instantiating the security kernel, the security kernel determining that the chooser value corresponds to the chooser application and therefore authenticating same, the chooser application being instantiated;

entering a normal mode after the chooser application is instantiated and leaving same to run, the chooser application presenting the plurality of available applications for selection by a user;

receiving a selection of one of the presented applications to be instantiated;

setting the chooser value to a value corresponding to the selected application;

entering a preferred mode upon an executed CPU reset and instantiating the security kernel, the security kernel determining that the chooser value corresponds to the selected application and therefore authenticating same, the selected application being instantiated;

entering a normal mode after the selected application is instantiated and leaving same to run;

wherein the security kernel allows the processor to be trusted to keep hidden a secret of the chooser application and a secret of the selected application.

42. (Original) The medium of claim 41 wherein the method further comprises setting the chooser value to the value corresponding to the chooser application upon the selected application being authenticated by the security kernel, wherein upon execution of a CPU reset, the security kernel determines that the chooser value corresponds to the chooser application 72c and therefore authenticates same.

43. (Original) The medium of claim 41 wherein the method further comprises storing the chooser value in a memory location not affected by a CPU reset so that the stored chooser value is available after same.

44-46. (Canceled)